Comparison of Multiclass Classification Algorithms for Music Recommendation
(ANN Course Project)

Gopal Sharma
gopalsharma1193@gmail.com
Enroll no: 12115043

Vibhor Goel
vibhoriitr@gmail.com
Enroll no: 12115120

1.Abstract

Music recommendation is receiving increasing attention as the music industry develops venues to deliver music over the Internet. The goal of music recommendation is to present user's lists of songs that they are likely to enjoy. Collaborative-filtering and content-based recommendations are two widely used approaches that have been proposed for music recommendation. However, collaborative filtering requires large amount of user preference dataset and thus makes this method difficult to implement in the starting of recommender system. Analyzing music audio files based on genres and other qualitative tags is an active field of research in machine learning. When paired with particular classification algorithms, most notably multi class support vector machines (SVMs) and k-nearest-neighbor classifiers (KNNs), certain features, including Mel-Frequency Cepstral Coefficients (MFCCs), Chroma attributes and other spectral properties, have been shown to be effective features for classifying music by genre. In this paper we apply various multiclass classification algorithms total of 32 acoustic features across one dataset, namely GTZAN. GTZAN dataset is professionally curated with features track-by-track basis. It originally has a thousand songs with hundred songs for each track. In this paper we will be comparing multiclass classification techniques like, one-vs-all, one-vs-one, DAG-SVM, KNN, multiclass neural network, Linear Discriminative analysis (LDA) and Random Forest on the 32 features collected on the GTZAN dataset.

2. Introduction:

Recommending music to listeners is a difficult problem that many systems (e.g., Pandora, Last.fm, Audiobaba, and Mystrands) attempt to solve. In these, user would first specify the genre or class of the music, and the algorithm would select music corresponding to the specified class (usually by music similarity). Some researchers argue that machine learning for music genre classification is not reliable for normal use because genres are not clearly defined. On the other hand, users using online music services are very likely to search for music by genre, so understanding how to automatically classify music by genre would only be a practical use. Nevertheless overarching genres like rock or classical likely exhibit enough distinction for machine learning algorithms to effectively distinguish between them. Hence many scientists have attempted—and largely succeeded—in producing quality classifiers for determining genres using better multiclass classification algorithms. In this effort, some papers have explored different learning and feature extraction techniques. In the paper George Tzanetakis effectively classified genres on live radio broadcasts using a Gaussian classifier [1]. Mandel used SVMs on artist and album-level features to make similar classifications as well [2]. One of the study explored mixtures of

Gaussians and k-nearest-neighbors for the same task [3]. All of these studies are using -Frequency Cepstral Coefficients and Chroma properties of the audio waveform, for instance—to make these classifications. Another research [8] used dynamic K-mean clustering, this algorithm clusters the pieces in the music list dynamically adapting the number of clusters. Recommendation is made using pieces of music based on the clusters.

2.1 DATASET:

The GTZAN dataset used and created by G. Tzanetakis and P. Cook and also used in their research of music classification. It has thousand songs, labeled by experts containing ten genres (each genre has hundred songs) and are collected from a variety of sources including personal CDs, radio, microphone recordings, in order to represent a variety of recording conditions. Although some of researches have shown faults in the dataset but in the present study, all the faults and miss-labeling have been ignored. All the tracks are 22055 Hz Mono 16-bit audio files in .wav format.

2.2 Features:

Once a set of features is obtained for a given signal, we can use supervised, semi-supervised or unsupervised learning methods to train the classification and retrieval engines on these features. Hence, stress is laid on extracting as many features as possible that can further relax the constraints on the classification procedure. A typical set of features for audio signals includes tonality, pitch (perceived fundamental frequency), temporal energy, harmonicity, timbre, spectral centroid, bandwidth and the Mel-Frequency Cepstral Coefficients (MFCC) etc. Features characterizing any audio signal can be majorly classified into two main categories – global descriptors and instantaneous descriptors. In the global, the feature is calculated for the entire signal as a whole. Such features help exploit stationary patterns of the signal such as the overall energy density of an audio clip or the emotional nature of a song. The latter features explain the short time nature and temporal evolution of a signal. These features can be calculated by segmenting the given signal into a set of overlapping or no overlapping frames and then applying the preprocessing techniques over each frame. In this paper, we are extracting the instantaneous descriptors for each frame. This category contains features that are mostly related to the temporal, spectral shape, harmonic and energy features. A brief description of the features and their extraction algorithms implemented in this paper are provided here and the flow diagram is shown below.

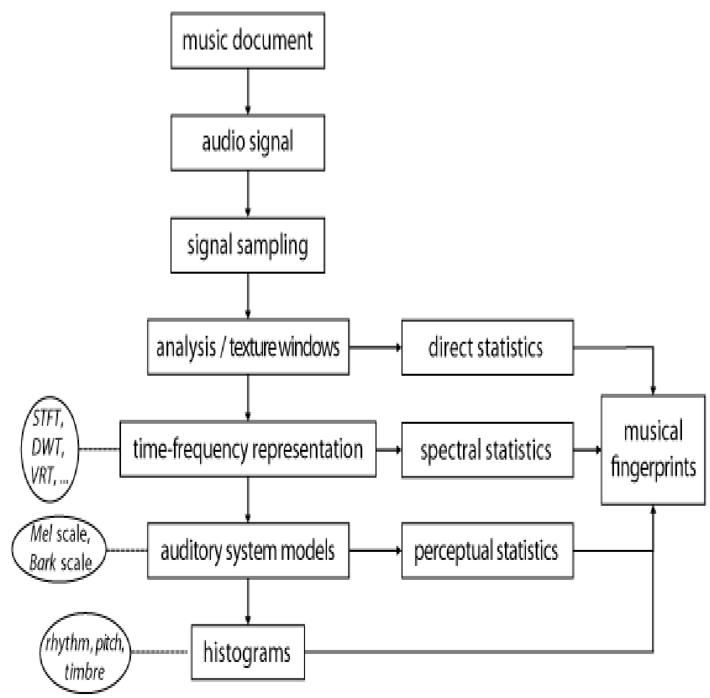

Fig 1. Flow diagram for feature extraction from songs.

I. Pitch

Pitch represents the periodicity inherent in the temporal domain or the perceived fundamental frequency of the underlying signal. Although the actual frequency can be determined accurately, it may differ from the pitch due to the presence of harmonics.

II. Temporal Energy

The temporal energy E can be computed by averaging the squared signal values over an entire frame.

III. Tonality Measure

Due to the presence of a large amount of background or sensor noise in an audio or speech signal, the original tone of the signal gets masked most of the times. Tonality is a measure of the signal's tone-like or noise-like characteristic. The Spectral Flatness Measure (SFM), defined as the ratio of the geometric mean (GM) to the arithmetic (AM) mean of the power spectrum P.A tonality value close to 1 indicates the presence of strong tonal components while 0 indicates noise-like signal as seen in Figure 4. Tonality plays an important role in perceptual coding of audio signals [14]. It is employed in psycho-acoustic models such as the MPEG.

IV. Spectral Centroid

Spectral centroid SC is defined as the mean of the distribution of frequency components for a given frame of the signal according to (6). The linear frequency or the Bark-scale can be used as parameters on which the weights (magnitude of FFT components) are applied. It gives an idea of the distribution of the frequency spectrum for a given signal. If the centroid is located towards the higher end of the spectrum, the signal exhibits a bright and happy sound and if it is located towards the lower end, it conveys a dull and gloomy sound.

V. Harmonicity:

Harmonicity features comprise of harmonicity ratio and the fundamental frequency. The former represents the degree to which periodicity is present in the signal in the probabilistic sense. The latter is the frequency, the multiple integer of which, best explains the content of the signal spectrum. The fundamental frequency is computed using a likelihood approximation based on Goldstein's algorithm [4]. The harmonicity value obtained for each frame conveys very little information. Hence, we average the harmonicity over a given number of frames. Harmonicity becomes an important feature for classification since it is large for speech, music and certain machine noises and smaller for most other types of environmental audio.

VI. Mel-Frequency Cepstral Coefficients

The MFCC represents the shape of the spectrum with very few coefficients. The cepstrum is defined as the Fourier transform of the logarithm of the spectrum. The Mel-Cepstrum is the spectrum computed on the Mel-bands instead of the Fourier spectrum. The use of Mel-cepstrum allows us to better account for the mid-frequency part of the signal.

2.3 Feature Extraction:

A MP3 file is converted into WAV. A 30 seconds audio file stored in WAV format which is passed to a feature extraction process. The WAV format for audio is simply the right and left stereo signal samples. The feature extraction process calculates 32 numerical features that characterize the particular sample. One of the features is MFCC that again gives 12 values. Feature extraction process is carried out on many different WAV files to create a matrix of containing columns of feature vectors. Feature extraction matrix is used to train neural network.

We used MARSYAS [5] to extract a number of relevant features from the raw audio files. When taken together, these features denote a wide

range of sonic characteristics of the music, including instrumentation, tonal variation, timbre texture, and production attributes. Together, they constitute the “musical surface” of the song [1]. Aim of our study is to compare existing multiclass algorithms to classify music into different genres. We have use support vector machines (SVM) which has various versions for multiclass classification, that are, DAG-SVM, one-vs-all, one-vs-one. We are also using Random Forest, Linear Discriminative Analysis (LDA), K-Nearest Neighbor (KNN) and Neural Network. We are describing some of them here:

2.4 Support Vector Machine:

These are supervised learning models with associated learning algorithms that analyze data and recognize models used for classification and regression analysis [9]. The basic SVM takes a set of input data and predicts, for each given input, which of two possible classes forms the output. Given the set of training examples each marked as belonging to two categories and SVM training algorithm builds some model that assigns the new example into one category or another.

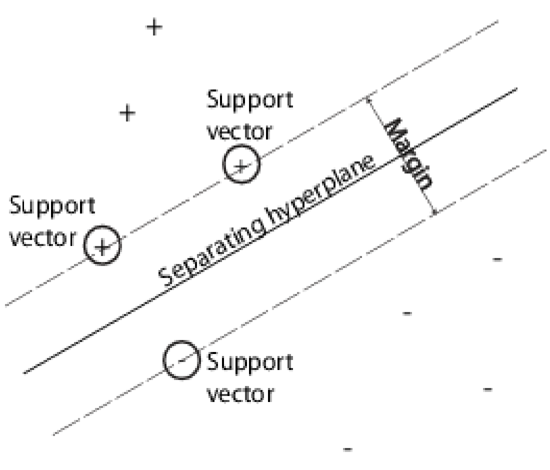

Fig 2. Figure representing separating hyper plane, support vectors, and stability margin.

I. ONE-AGAINST-ALL

The earliest used implementation for SVM multiclass classification is probably the one-against-all method. It constructs k SVM models where k is the number of classes. The ith SVM is trained with all of the examples in the ith class with positive labels, and all other examples with negative labels.

Thus the given l training data $(x_1, y_1) \ldots . (x_l, y_l)$, where

$$x_i \in R^n, i = 1 \ldots., l$$
$$y_i \in \{1, \ldots\ldots, k\}$$

Is the class of x_i, the ith SVM solves the dual problem where the training data x_i are mapped to a higher dimensional space by the function $\oint$ and C is the penalty parameter.

Minimizing $(1/2)(w^i)^T w^i$ means that we would like to maximize$2/\| w \|$, the margin between two groups of data. When data are not linear separable, there is a penalty term $C\sum_{j=1}^{l} \xi_j^i$ which can reduce the number of training errors. The basic 3 concept behind SVM is to search for a balance between the regularization term $(1/2)(w^i)^T w^i$ and the training errors.

Class of $x = \arg max\ x_{i=1,\ldots,k}\ (\left(w^i\right)^T \oint \left(x_j\right) + b_j)$

Practically, we solve the dual problem whose number of variable is the same as the number of data. Hence k l-variable quadratic programming problems are solved. We used LIBSVM [4] to solve the support vector machine problem. In this article, we have used Radial Basis kernel in Libsvm. We have used the cost value 10 for error margin and we are doing probability estimation also through svmpredict matlab function which gives us the predicted label, accuracy estimates and maximum margin probability for each test data using which we obtain accuracy of each genre.

II. ONE-AGAINST-ONE

Another major method is called the one-against-one method. It was introduces in [6], and the first use of this strategy on SVM was in [5]. This method constructs $\frac{k(k-1)}{2}$ classifiers where each one is trained on data from two classes. For training data from the $ith\ and\ jth$ classes, we solve the binary classification problem of SVM. There are different methods for doing the future testing after all $\frac{k(k-1)}{2}$ classifiers are constructed. After some tests, we decide to use the following voting strategy suggested in [6]. If $((w^{ij})^T \oint (x_j) + b_{ij})$ says x is in the ith class, then the vote for the ith class is added by one. Otherwise, the voting for the jth class is added by one. Then we predict x is in the class with the largest vote. The voting approach described above is also called the “Max Wins” strategy. In case that two class have the same votes, though it may not be a good strategy, now we simply select the one with the smaller index. Practically we solve the dual of (3) whose number of variables is the same as the number of data in two classes. Hence if in average each class have l data points, we have to solve $\frac{k(k-1)}{2}$ quadratic programming problems where each of them has about 2 l / k variables.

III. DAG SVM

The last algorithm discussed here is the directed acyclic graph SVM (DAGSVM) proposed in [7]. Its training phase is the same as the one-against-one method by existence $\frac{k(k-1)}{2}$ binary SVMs. However, in the testing phase, it uses a rooted binary directed acyclic graph which has existence $\frac{k(k-1)}{2}$ internal nodes and k leaves. Each node is a binary SVM of ith and jth classes. Given a test sample x, starting at the root node, the binary decision function is evaluated. Then it moves to either left or right depending on the output value. Therefore, we go through a path before reaching a leaf node which indicates the predicted class. ABC vs DE A vs BC D vs E A B vs C D E B C 5 The figure below shows us an example of DAGSVM (A, B, C, D, E represents five classes).

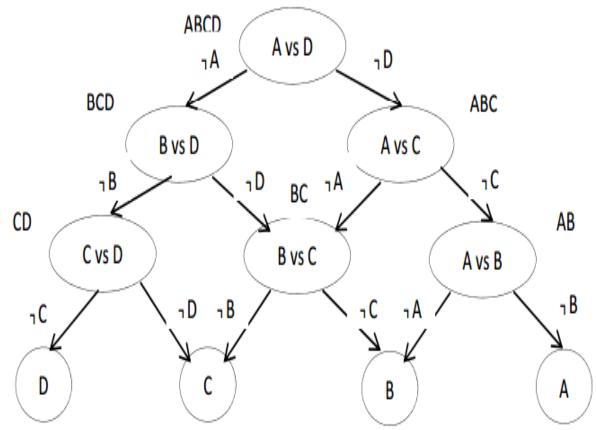

Fig 3. Decision classification tree used in DAGSVM.

3. Results:

I. SVM Classification:

In this research paper, we are using three methods of comparing Support Vector machine classification. All three are widely used:

a) One versus one classification:

In this method, we train the data taking two classes at a time. Thus, having $\frac{k(k-1)}{2}$ training cycles where k is the number of genres. This method is long and comparatively less accurate.

On the GTZAN dataset, using one versus one classification, we got an accuracy of 75.625% taking first 8 classes. On class basis, we see that classical genre has very good accuracy while blues genre has poor accuracy of only 60%.

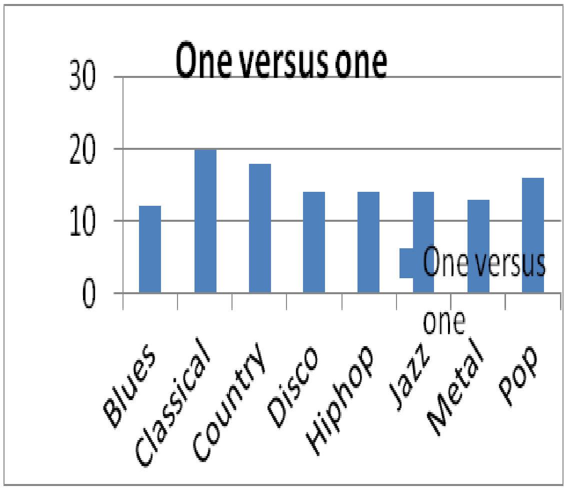

Fig 4. Genre wise correct predictions on testing 20 songs for each genre using One versus one classification

b) One versus all classification:

In this method we, compared the songs taking one genre as one class and all other genres as another class. We repeated this for all the classes and thus had k training cycles where k is the number of genres. The best class was chosen on basis of maximum margin probability. We obtained best accuracy for one versus all classification over DAGSVM and one versus one classification.

We obtained an accuracy of 77.5 percent. For one versus all classification, the disco genre performed worst giving accuracy of only 60 percent.

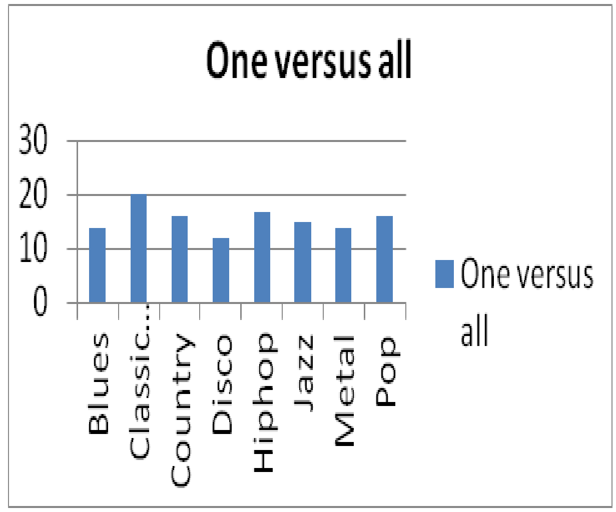

Fig 5. Genre wise correct predictions on testing 20 songs for each genre using One versus all classification.

c) DAGSVM:

DAGSVM uses directed acyclic graph for classification. It is an improvement over one versus one classification taking less number of training cycles.

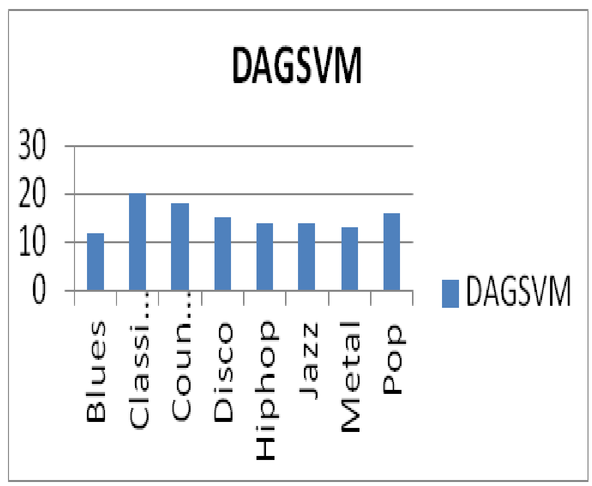

Fig 6. Genre wise correct predictions on testing 20 songs for each genre using DAGSVM classification.

The results obtained from DAGSVM were quite similar to one versus one classification but only little better with total accuracy of 76.25 percent.

II. Random Forest:

Random forest uses the concept of decision tree classification and randomization. In the earlier papers reviewed in this article, none of them have employed random forest for classification. We have gone a step further by using the information from different trees obtained to classify the data. We obtained the results as shown in the figure, we see that genres "disco" and "jazz" have unusually low accuracy for random forest algorithm. On an overall basis, the random forest algorithm gives an accuracy of 63.12 percent.

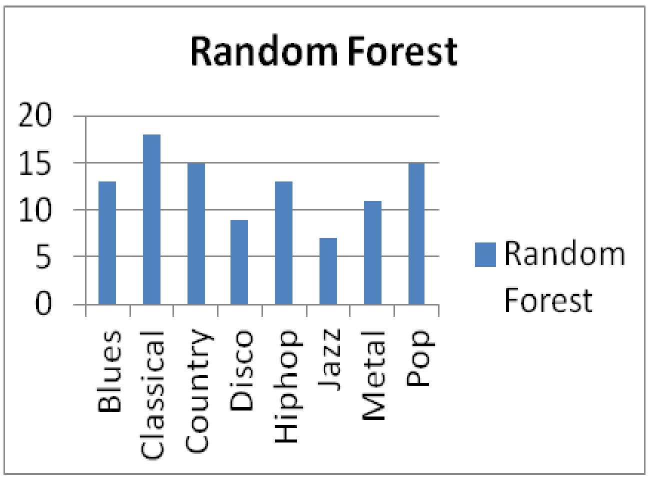

Fig 7. Genre wise correct predictions on testing 20 songs for each genre using RandomForest classification.

III. Least Discriminant analysis:

We use linear discriminant analysis for classification. On an overall basis, it performs better than SVM one versus one classification while performing poor than one versus all and DAGSVM classification. We obtained an overall accuracy of 65.67 percent.

IV. K nearest Neighbors:

In K nearest neighbor each data point is given an average class of its k nearest data points. This method is based on distance metrics and hence we expect comparatively less accuracy than training and testing based methods.

We obtained an accuracy of 49.5 percent using K nearest neighbor algorithm.

V. Neural Networks:

Neural networks can solve some really interesting problems once they are trained. They are very good at pattern recognition problems and with enough elements (called neurons) can classify any data with arbitrary accuracy. They are particularly well suited for complex decision boundary problems over many variables. Here, we classify the data set using neural network in matlab.

4. Conclusion:

We see that K nearest neighbor algorithm performs poorly for classification. We obtained good accuracy using SVM classification. For one versus one classification, the training process is slow and grows in square proportion as number of classes is increased. Hence it is not a suitable method for classification.

In one-versus all classification accuracy obtained for this data set is good but it is advised to avoid this because of large ratio between number of members to belonging to each class and total members as the number of classes is increased.

Hence, it is better to use DAGSVM for classification. It uses less training time and both the sets have equal ratio in each training cycle. Use of random forest can also be done, but it suffers from inherent randomization which reduces the accuracy of classification. Using LDA approach, we see that the accuracy is normal and we also do not face problems like different ratio between numbers of members in each classification.

5. References:

[1] Tzanetakis, George, Georg Essl, and Perry Cook. "Automatic Musical Genre Classification Of Audio Signals." The International Society for Music Information Retrieval. 2001. http://ismir2001.ismir.net/ pdf/tzanetakis.pdf

[2] Mandel, Michael I. and Daniel P.W. Ellis. “Song-level features and support vector Machines for music classification”. The International Society for Music Information Retrieval. Columbia University, 2005. http://www.ee.columbia.edu/~dpwe/pubs/ismir05-svm.pdf

[3] Li, Tao, Mitsunori Ogihara, and Qi Li. “A Comparative Study on Content-Based Music Genre Classification”. Special Interest Group on Information Retrieval. 2003, p. 282. http://users.cis.fiu.edu/~taoli/pub/sigir03-p282-li.pdf

[4]Chang, Chih-Chung and Lin, Chih-Jen, LIBSVM

“ACM Transactions on Intelligent Systems and Technology”,2,3,2011,pages:27:1—27:27, software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm/.

[5] S. Knerr, L. Personnaz, and G. Dreyfus, “Single-layer learning revisited: A step- wise procedure for building and training a neural network,” in Neurocomputing: Alogrithms, Architectures and Applications, J. Fogelman, Ed. New York: Spri-nger-Verlag, 1990.

[6] Kerebel U. Pairwise Classification and Support Vector Machine. In: Scholkopf B, Burges, C, Smola A, eds. Advances in Kernel Methods-Support Vectoring.Com- bridge, MA: MIT Press, 1999: 255-268.

[7] Platt J C, Chrisianini N, Shawe-Taylor J. Large Margin DAGs for Multiclass Classification. In Advances in Neural Information Processing Systems12. Combri- ge, MA: MIT Press, 2000: 547-553.

[8] D. Kim, K. Kim, K. Park, J. Lee, and K. Lee, “A music recommendation system with a dynamic k-means clustering algorithm,” in Proceedings of the 6th International Conference on Machine Learning and Applications, 2007, pp. 399-403.

[9]C.J.C. Burges, ‘A tutorial on support vector machines for pattern recognition’, Data Mining and Knowledge Discovery, 2 (2) (1998), pp. 121–167.